Credit Card Fraud Detection: Approach and Methodology

1. Introduction

The goal of this project is to predict fraudulent credit card transactions using machine learning techniques. The dataset, sourced from Kaggle, contains 284,807 transactions, of which only 492 are fraudulent. This significant class imbalance poses a central challenge to model building and evaluation.

2. Business Problem Overview

Fraudulent transactions pose severe risks to banks and customers, leading to financial losses, reduced trust, and reputational damage. As digital payment channels grow, so do opportunities for fraud, making proactive detection and prevention essential. Machine learning provides scalable solutions to detect fraud efficiently, reducing manual reviews and chargebacks while safeguarding legitimate transactions.

3. Understanding and Defining Fraud

Credit card fraud encompasses unauthorized actions to gain financial benefit, such as skimming, card manipulation, creation of counterfeit cards, theft, and fraudulent telemarketing. Effective detection systems must be robust against these diverse methods.

4. Data Understanding

The dataset includes anonymized features (V1–V28) derived from PCA, along with 'Time', 'Amount', and 'Class' labels. The 'Class' variable indicates fraud (1) or non-fraud (0). The minority class (fraud) makes up only 0.172% of the data, requiring special handling during modeling.

5. Approach Overview

Step 1: Data Understanding

- Loaded and reviewed the dataset, focusing on the distribution of features and the class imbalance.
- Confirmed that principal components (V1–V28) do not require further scaling.

Step 2: Exploratory Data Analysis (EDA)

- Performed univariate and bivariate analyses to understand the relationship of features with the target variable.
- Checked for skewness in 'Amount' and 'Time' and applied transformations if necessary.
- Assessed the severe class imbalance and visualized the distribution of fraudulent vs. non-fraudulent transactions.

Step 3: Handling Class Imbalance

- **Applied oversampling techniques (ADASYN and SMOTE)** to balance the minority (fraud) class.
- Evaluated the effect of oversampling on class distribution and model performance.

Step 4: Train/Test Split and Cross-Validation

- Split the data into training and testing sets, ensuring representative distribution of the minority class.
- Used **StratifiedKFold cross-validation** to maintain class proportion in each fold.
- This approach ensures robust model assessment, given the class imbalance.

Step 5: Model Building and Hyperparameter Tuning

- Built and compared multiple models:
    - Logistic Regression (L1 and L2 Regularization)
    - K-Nearest Neighbors (KNN)
    - Decision Trees (Gini and Entropy)
    - Random Forest
    - XGBoost
- Focused on **XGBoost** due to its superior performance on oversampled data.
- Tuned hyperparameters (max_depth, min_child_weight, n_estimators, learning_rate, gamma, subsample, colsample_bytree) using GridSearchCV.

Step 6: Model Evaluation

- Evaluated models primarily using **ROC AUC**, as it is more sensitive to the minority class and better reflects the business goal of identifying fraud.
- Also considered accuracy, but prioritized metrics that emphasize correct identification of fraudulent transactions.
- Determined the optimal probability threshold for classification using Youden's J statistic (maximizing TPR - FPR).

6. Results & Model Selection

- **XGBoost with Random Oversampling and StratifiedKFold CV** delivered the best results:
    - Model Accuracy: 0.9994
    - ROC AUC: 0.9933
    - Optimal Threshold: 0.8651
- Logistic Regression with L2 regularization and RepeatedKFold CV was the best performer without oversampling, but was outperformed by XGBoost on balanced data.

7. Conclusion & Recommendation

- The chosen approach—**XGBoost trained on Random Oversampled data with StratifiedKFold cross-validation**—proved most effective for this highly imbalanced classification problem.
- This methodology ensures high detection rates for fraudulent transactions while minimizing false alarms.
- Further hyperparameter tuning and ensemble methods can be explored for marginal gains, but the current approach is robust and production-ready.